Tools for Machine Learning:

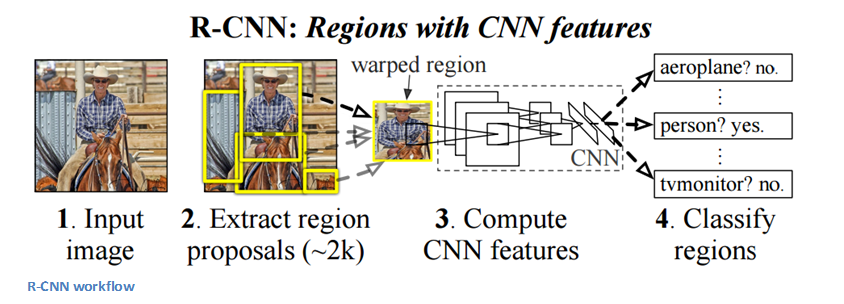
# Region Based CNNs (2013, 2015)

### [R-CNN](https://arxiv.org/pdf/1311.2524v5.pdf) - 2013, [Fast R-CNN](https://arxiv.org/pdf/1504.08083.pdf) - 2015, [Faster R-CNN](http://arxiv.org/pdf/1506.01497v3.pdf) - 2015

Some may argue that the advent of R-CNNs has been more impactful that any of the previous papers on new network architectures. With the first R-CNN paper being cited over 1600 times, Ross Girshick and his group at UC Berkeley created one of the most impactful advancements in computer vision. As evident by their titles, Fast R-CNN and Faster R-CNN worked to make the model faster and better suited for modern object detection tasks.

The purpose of R-CNNs is to solve the problem of object detection. Given a certain image, we want to be able to draw bounding boxes over all of the objects. The process can be split into two general components, the region proposal step and the classification step.

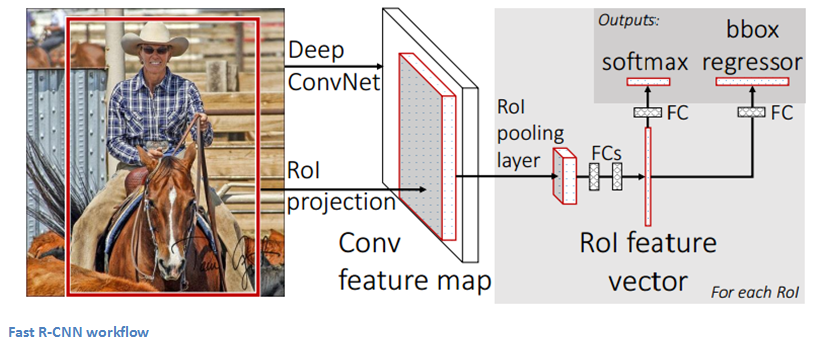
The authors note that any class agnostic region proposal method should fit. [Selective Search](https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013/UijlingsIJCV2013.pdf) is used in particular for RCNN. Selective Search performs the function of generating 2000 different regions that have the highest probability of containing an object. After we’ve come up with a set of region proposals, these proposals are then “warped” into an image size that can be fed into a trained CNN (AlexNet in this case) that extracts a feature vector for each region. This vector is then used as the input to a set of linear SVMs that are trained for each class and output a classification. The vector also gets fed into a bounding box regressor to obtain the most accurate coordinates.



Non-maxima suppression is then used to suppress bounding boxes that have a significant overlap with each other.

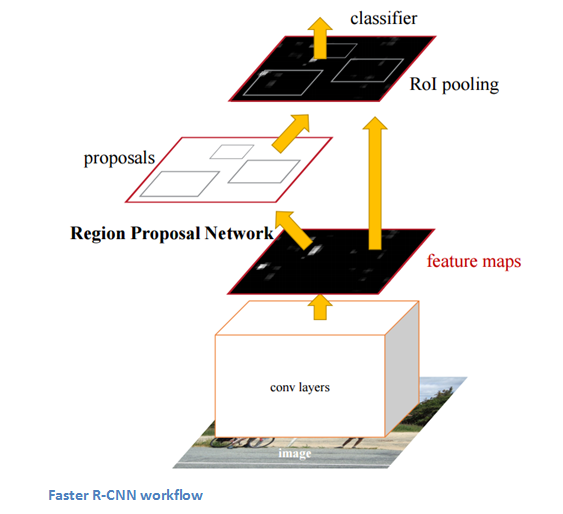
**Fast R-CNN**

Improvements were made to the original model because of 3 main problems. Training took multiple stages (ConvNets to SVMs to bounding box regressors), was computationally expensive, and was extremely slow (RCNN took 53 seconds per image). Fast R-CNN was able to solve the problem of speed by basically sharing computation of the conv layers between different proposals and swapping the order of generating region proposals and running the CNN. In this model, the image is *first*fed through a ConvNet, features of the region proposals are obtained from the last feature map of the ConvNet (check section 2.1 of the [paper](https://arxiv.org/pdf/1504.08083.pdf) for more details), and lastly we have our fully connected layers as well as our regression and classification heads.



**Faster R-CNN**

Faster R-CNN works to combat the somewhat complex training pipeline that both R-CNN and Fast R-CNN exhibited. The authors insert a region proposal network (RPN) after the last convolutional layer. This network is able to just look at the last convolutional feature map and produce region proposals from that. From that stage, the same pipeline as R-CNN is used (ROI pooling, FC, and then classification and regression heads).



**Why It’s Important**

Being able to determine that a specific object is in an image is one thing, but being able to determine that object’s exact location is a huge jump in knowledge for the computer. Faster R-CNN has become the standard for object detection programs today.